Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®
Discrete Applied Mathematics 136 (2004) 3–11
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

$(2 + f(n))$-SAT and its properties☆

Yunlei Zhao[a,b,*], Xiaotie Deng[a], C.H. Lee[a], Hong Zhu[b]

[a] *Department of Computer Science, City University of Hong Kong, Hong Kong, China*
[b] *Department of Computer Science, Fudan University, Shanghai, China*

Received 25 July 2000; received in revised form 10 January 2002; accepted 26 August 2002

Abstract

Consider a formula that contains n variables with the form $\Phi = \Phi_2 \wedge \Phi_3$, where Φ_2 is an instance of 2-SAT containing m_2 2-clauses and Φ_3 is an instance of 3-SAT containing m_3 3-clauses. Φ is an instance of $(2 + f(n))$-SAT if $m_3/(m_2 + m_3) \leqslant f(n)$. We prove that $(2 + f(n))$-SAT is in $\mathscr{P}$ if $f(n) = O(\log n/n^2)$, and in $\mathscr{NPC}$ if $f(n) = 1/n^{2-\varepsilon}(\forall \varepsilon\colon 0 < \varepsilon < 2)$. Most interestingly, we give a candidate, $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$, for natural problems in $\mathscr{NP} - \mathscr{NPC} - \mathscr{P}$ (denoted as $\mathscr{NPI}$) with respect to this $(2 + f(n))$-SAT model. We prove that the restricted version of it is not in $\mathscr{NPC}$ under $\mathscr{P} \neq \mathscr{NP}$. Actually, it is indeed in $\mathscr{NPI}$ under some stronger but plausible assumption, specifically, the exponential-time hypothesis.

Keywords: Computational complexity; SAT; Exponential-time hypothesis

1. Introduction

In 1975, Lander had shown that there exist some languages in $\mathscr{NP}$–$\mathscr{NPC}$–$\mathscr{P}$ (denoted as $\mathscr{NPI}$) under the assumption $\mathscr{P} \neq \mathscr{NP}$ [8]. But the language constructed there is not a natural one because the construction needs to run all Turing machines. So far, no natural problems have been proven to be in $\mathscr{NPI}$ under $\mathscr{P} \neq \mathscr{NP}$ and finding such a natural problem is considered an important open problem in complexity theory [13,4]. The problems of graph isomorphism GI and factoring, which were suggested by Karp, are regarded as two most likely candidates [13,4].

☆ This research is supported by a research grant of City University of Hong Kong 7001023.
* Corresponding author. Department of Computer Science, City University of Hong Kong, Hong Kong, China.
E-mail addresses: csylzhao@cityu.edu.hk (Y. Zhao), csdeng@cityu.edu.hk (X. Deng), cschlee@cityu.edu.hk (C.H. Lee), hzhu@fudan.edu.cn (Hong Zhu).

0166-218X/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0166-218X(03)00194-X

The satisfiability problem of Boolean formula (SAT) has played a central role in the field of computational complexity theory. It is the first $\mathcal{NP}$-complete problem. And up to now, all known algorithms to find a solution for 3-SAT require exponential time in problem size in the worst case. In practice, the time complexity of the fastest algorithm for 3-SAT is $(\frac{4}{3})^n$, where n is the variable number in the formula [14]. It is also an important open question whether sub-exponential time algorithms exist. The plausibility of such a sub-exponential time algorithm for 3-SAT was investigated in [5], using sub-exponential time reduction. It was shown there that linear size 3-SAT is complete for the class $\mathcal{SNP}$ (strict $\mathcal{NP}$) with respect to such reduction. It implies that if there exists a sub-exponential time algorithm for 3-SAT then all the languages in $\mathcal{SNP}$ can be decided in sub-exponential time. Note that some well-studied problems, such as k-SAT, k-colorability, for any $k \geqslant 3$, and so on, have been proven to be $\mathcal{SNP}$-complete. In light of both the practical and theoretical supports, Impagliazzo and Paturi introduced the exponential-time hypothesis (ETH) for 3-SAT: 3-SAT does not have a sub-exponential-time algorithm [6]. Although ETH is stronger than $\mathcal{NP} \neq \mathcal{P}$, it is still quite reasonable. In recent advances of cryptography, many important cryptographic primitives and protocols were constructed under the ETH for the one-way functions: DLP or RSA, e.g., verifiable pseudorandom functions [9], verifiable pseudorandom generator [3] and resettable zero-knowledge arguments systems for $\mathcal{NP}$ [2,10] and so on.

On the other hand, recently there has been growth of interests to study the link between the hardness of computational complexity of decision problems and the phase boundaries in physical systems [1,12]. It was observed that, similar to physical systems, across certain phase boundaries dramatic changes occur in the computational difficulty and solution character. $\mathcal{NP}$-complete problems become easier to solve away from the boundary and the hardest problems occur at the phase boundary [7,12].

To understand the onset of exponential complexity that occurs when going from a problem in $\mathcal{P}$(2-SAT) to a problem that is $\mathcal{NP}$-complete (3-SAT), the $(2+p)$-SAT model was introduced in [11,12], where p is a constant and $0 \leqslant p \leqslant 1$. An instance of $2+p$-SAT is a formula with m clauses, of which $(1-p)m$ contain two variables (2-clauses) and pm contain three variables (3-clause). $2+p$-SAT smoothly interpolates between 2-SAT ($p=0$) and 3-SAT ($p=1$) when the instances are generated randomly. The median computation cost scales linearly with n (the number of variables) when $p < p_0$ and exponentially for $p > p_0$, where p_0 lies between 0.4 and 0.416 [12]. However, for the worst-case complexity, $(2+p)$-SAT is $\mathcal{NP}$-complete for any constant p, $p > 0$ [12,1].

In this work, we further explore the worst-case complexity boundary of $\mathcal{P}$ and $\mathcal{NPC}$ when p is further reduced (not a constant but a function of n). Somewhat surprisingly, such an extension allows us to suggest another candidate for natural problems in $\mathcal{NPI}$ under $\mathcal{NP} \neq \mathcal{P}$. In fact, we present a natural problem in $\mathcal{NPI}$ under ETH. In Section 2, we present the necessary definitions and the related important properties for our study. In Section 3, we present a candidate for natural problems in $\mathcal{NPI}$ and prove it not in $\mathcal{NPC}$ under $\mathcal{NP} \neq \mathcal{P}$. In Section 4, we prove it is not in $\mathcal{P}$ under ETH. We conclude with discussions in Section 5.

2. Properties of $(2 + f(n))$-SAT

In this section, we introduce the $(2 + f(n))$-SAT model. We are mainly concerned with the boundary of $f(n)$ that separates the problems between $\mathscr{P}$ and $\mathscr{NPC}$.

Let Φ is an formula and denoted $|\Phi|$ as the number of clauses in Φ. We introduce the definition of $(2 + f(n))$-SAT:

Definition 2.1 ($(2+f(n))$-SAT). Consider a formula which contains n variables and m clauses with the form $\Phi = \Phi_2 \wedge \Phi_3$, where Φ_2 is an instance of 2-SAT which contains m_2 2-clauses, and Φ_3 is an instance of 3-SAT which contains m_3 3-clauses. An instance of $(2 + f(n))$-SAT is one satisfying the condition

$$\frac{|\Phi_3|}{|\Phi|} = \frac{m_3}{m} = \frac{m_3}{m_2 + m_3} \leqslant f(n).$$

Throughout the paper, we restrict our discussion to instances with $f(n) = |\Phi_3|/|\Phi|$. Indeed, all our claims hold if they hold under this restriction. Note that $m_2 \leqslant 4n_2^2$, $m_3 \leqslant 8n_3^3$, $n_2 \leqslant 2m_2$, $n_3 \leqslant 3m_3$, $n \leqslant 3m$, and that the variables which appear in Φ_2 may appear in Φ_3, and vice versa, i. e., $n \leqslant n_2 + n_3 \leqslant 2n$.

Theorem 2.1. *For any constant $k > 0$, $(2 + k \log n/n^2)$-SAT is in $\mathscr{P}$.*

Proof. Consider any instance of $(2 + k \log n/n^2)$-SAT ($k > 0$), a formula $\Phi = \Phi_2 \wedge \Phi_3$, where $m_3/(m_2 + m_3) = k \log n/n^2$. We get

$$m_3 = \frac{k \log n m_2}{n^2 - k \log n} \leqslant \frac{k m_2 \log n + k \log n}{n^2} \leqslant \frac{(k4n^2 + k) \log n}{n^2}$$

$$= \left(4k + \frac{k}{n^2}\right) \log n \leqslant 5k \log n.$$

Note that the variables which appear in Φ_2 may appear in Φ_3, and vice versa. For the $5k \log n$ variables which appear in Φ_3, we can enumerate all the at most n^{5k} truth assignments and then for each truth assignment we can determine Φ_2 in polynomial time of n, and thus the $(2 + k \log n/n^2)$-SAT ($k \geqslant 0$) is in $\mathscr{P}$. □

Claim 1. *Given n variables, we can construct a satisfiable formula Φ, where Φ is an instance of 2-SAT and $|\Phi| \leqslant \frac{3}{2}n^2 - \frac{3}{2}n$.*

Proof. We construct 2-clauses as follows: $(\frac{1}{2}n^2 - \frac{1}{2}n)$ clauses with the form $(x_i \vee x_j)(i \neq j, 1 \leqslant i, j \leqslant n)$, $(n^2 - n)$ clauses with the form $(x_i \vee \neg x_j)$, $(i \neq j, 1 \leqslant i, j \leqslant n)$. From all these 2-clauses, we select $k, 1 \leqslant k \leqslant \frac{3}{2}n^2 - \frac{3}{2}n$, clauses to construct the formula Φ we need, then Φ is satisfiable when all these n variable are assigned the value "true". □

Theorem 2.2. $(2 + \frac{1}{n^{2-\varepsilon}})$-*SAT* $(\forall \varepsilon,\ 0 < \varepsilon < 2)$ *is in* $\mathscr{NPC}$.

Proof. We show that there is a many-one reduction from 3-SAT to $(2+\frac{1}{n^{2-\varepsilon}})$-SAT $(0<\varepsilon<2)$. Let Φ_3 be an instance of 3-SAT that contains n_3 variables and m_3 3-clauses. Without loss of generality, we assume that $m_3 \geqslant 2$. Then we add $n_2 = m_3^{8/\varepsilon}$ new variables and using these new variables to construct a satisfiable formula Φ_2 which contains m_2 2-clauses.

Let $m_3/(m_2+m_3) = 1/n^{2-\varepsilon}(0<\varepsilon<2)$ then

$$\frac{m_3}{m_2+m_3} = \frac{1}{n^{2-\varepsilon}} \geqslant \frac{1}{(n_2+n_3)^{2-\varepsilon}},$$

$$m_2 \leqslant ((n_2+n_3)^{2-\varepsilon}-1)m_3 \leqslant (n_2+n_3)^{2-\varepsilon}m_3 \leqslant (m_3^{8/\varepsilon}+3m_3)^{2-\varepsilon}m_3.$$

But note that $m_3 \geqslant 2$, we get

$$\begin{aligned}(m_3^{8/\varepsilon}+3m_3)^2 m_3 &\leqslant \left[\frac{3}{2}(m_3^{8/\varepsilon})^2 - \frac{3}{2}m_3^{8/\varepsilon}\right](m_3)^8 \\ &= \left(\frac{3}{2}n_2^2 - \frac{3}{2}n_2\right)m_3^8 \\ &\leqslant \left(\frac{3}{2}n_2^2 - \frac{3}{2}n_2\right)(m_3^{8/\varepsilon}+3m_3)^{\varepsilon}.\end{aligned}$$

That is,

$$m_2 \leqslant (m_3^{8/\varepsilon}+3m_3)^{2-\varepsilon}m_3 \leqslant \frac{3}{2}n_2^2 - \frac{3}{2}n_2 \Rightarrow m_2 \leqslant \frac{3}{2}n_2^2 - \frac{3}{2}n_2.$$

The satisfiable formula Φ_2 can be constructed according to Claim 1.

Let $\Phi = \Phi_2 \wedge \Phi_3$, then Φ is an instance of $(2+1/n^{2-\varepsilon})$-SAT $(0<\varepsilon<2)$ and Φ is satisfiable if and only if Φ_3 is satisfiable.

Note that the above many-one reduction indeed can be constructed in polynomial time of m_3 (also in polynomial time of n_3, since $n_3 \leqslant 3m_3$, $m_3 \leqslant 8n_3^3$).

Obviously, $(2+1/n^{2-\varepsilon})$-SAT $(0<\varepsilon<2)$ is in $\mathcal{NP}$, so the theorem does hold. □

One open problem related to our $(2+f(n))$-SAT model is:

Open problem. Does there exist some $f(n)$, s.t. $k\log n/n^2 < f(n) < 1/n^{2-\varepsilon}$, where $k \geqslant 0$ and $0<\varepsilon<2$, so that $(2+f(n))$-SAT is in $(\mathcal{NP}$–$\mathcal{NPC})$–$\mathcal{P}$ (denoted as $\mathcal{NPI}$) under the assumption $\mathcal{P} \neq \mathcal{NP}$?

Note that $(2+k\log n/n^2)$-SAT is in $\mathcal{P}$, $k \geqslant 0$ and $(2+1/n^{2-\varepsilon})$-SAT $(0<\varepsilon<2)$ is in $\mathcal{NP}$-complete according to the above theorems.

Now, we give another candidate and also another open problem with regard to our $(2+f(n))$-SAT for natural problems in $\mathcal{NPI}$ under $\mathcal{P} \neq \mathcal{NP}$:

Open problem. In the $(2+f(n))$-SAT model, is $(2+(\log n)^k/n^2)$-SAT $(k \geqslant 2)$ in $(\mathcal{NP}$–$\mathcal{NPC})$–$\mathcal{P}$ under the assumption $\mathcal{NP} \neq \mathcal{P}$?

Note that $k_1\log n/n^2 < (\log n)^k/n^2 (k \geqslant 2) < 1/n^{2-\varepsilon}$, where $k_1 \geqslant 0$ and $0<\varepsilon<2$.

3. A candidate for natural problems in $\mathcal{NPI}$ under $\mathcal{NP} \neq \mathcal{P}$

Now, we give another candidate for natural problems in $\mathcal{NPI}$ under $\mathcal{P} \neq \mathcal{NP}$ which is a restricted version of $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$. We will prove that it is not $\mathcal{NP}$-complete under the assumption $\mathcal{P} \neq \mathcal{NP}$. Actually, it is indeed in $\mathcal{NPI}$ under some stronger but reasonable assumptions.

Theorem 3.1. *In the* $(2 + f(n))$*-SAT model, if the variables which appear in* Φ_2 *do not appear in* Φ_3*, and vice versa, then* $(2 + (\log n)^k/n^2)$*-SAT is not in* $\mathcal{NPC}$ *under the assumption* $\mathcal{NP} \neq \mathcal{P}$*,* $k \geqslant 2$*.*

Proof. Clearly, this problem is in $\mathcal{NP}$. We prove this theorem by showing that 3-SAT cannot be reduced to $(2 + (\log n)^k/n^2)$-SAT by many-one reduction, where $k \geqslant 2$.

Assume that there exists a many-one reduction(denoted as F) from 3-SAT to $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$. It means that for any instance of 3-SAT, a formula Φ_0 which contains n_0 variables and m_0 3-clauses, we can construct the $F(\Phi_0)$ which is an instance of $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$ in polynomial time of n_0, where $F(\Phi_0)$ contains n variables and m clauses, and $F(\Phi_0)$ is satisfiable if and only if Φ_0 is satisfiable. Let $F(\Phi_0) = \Phi_2 \wedge \Phi_3$, where Φ_2 is an instance of 2-SAT which contains m_2 2-clauses and n_2 variables and Φ_3 is an instance of 3-SAT which contains m_3 3-clauses and n_3 variables, then $(\log n)^k/n^2 = |\Phi_3|/|\Phi| = m_3/m = m_3/(m_2 + m_3)$, $k \geqslant 2$.

We consider the relation between m_3 and m_0 there are two cases:

Case 1: $m_3 \geqslant m_0$.

Claim 2. $m = m_2 + m_3$ *cannot be expressed as a polynomial of* m_3*.*

Proof (of Claim 2). Firstly, for sufficiently large n, $(\log n)^k/n^2 = m_3/m \leqslant \frac{1}{2}$ (i.e. $m \geqslant 2m_3$), where $k \geqslant 2$. Secondly,

$$m = m_2 + m_3 \leqslant 4n^2 + m_3 \Rightarrow n^2 \geqslant \frac{m - m_3}{4}.$$

Then, for sufficiently large n, the following holds:

$$\frac{m_3}{m} = \frac{(\log n)^k}{n^2} \leqslant \frac{4(\log 3m)^k}{m - m_3}$$

$$\Rightarrow 4(\log 3m)^k \geqslant m_3 \frac{m - m_3}{m} \geqslant \frac{1}{2} m_3$$

$$\Rightarrow m \geqslant \frac{1}{3} 2^{(\frac{m_3}{8})^{1/k}}. \quad \square$$

According to Claim 2, in Case 1, we get the fact that m cannot be expressed as a polynomial of m_3, and since $m_3 \geqslant m_0$, so m also cannot be expressed as a polynomial of m_0 (of course m also cannot be expressed as a polynomial of n_0 since $m_0 \leqslant 8n_0^3$). Its absurd since the many-one reduction $F(\Phi_0)$ must be done in polynomial time of n_0.

Case 2: $m_3 < m_0$. Since we assume $F(\Phi_0)$ can be constructed in polynomial time of n_0, then m_2 must be expressed as $P(n_0)$, where $P(\cdot)$ is a polynomial. So, if $m_3 < m_0$ it means that we can decrease the 3-clause number in Φ_0 by adding $P(n_0)$ 2-clauses (by imposing F on Φ_0). However, note that we assume the variables which appear in Φ_2 do not appear in Φ_3, and vice versa, then we can impose F on Φ_3, and so on. Repeat the above process at most m_0 times we can eliminate all 3-clauses in $F(\Phi_0)$ to get a formula Φ' and guarantee that Φ' is satisfiable if and only if $F(\Phi_0)$ is satisfiable if and only if Φ_0 is satisfiable, where Φ' contains only 2-clauses and $|\Phi'|$ is at most $m_0P(n_0)$, or at most $8n_0^3P(n_0)$, another polynomial of n_0. This means that there exists a many-one reduction from 3-SAT to 2-SAT, which contradicts our assumption $\mathscr{P} \neq \mathscr{NP}$.

So, from the arguments above, we can conclude that $(2+(\log n)^k/n^2)$-SAT $(k \geqslant 2)$ is not $\mathscr{NP}$-complete under the assumption $\mathscr{P} \neq \mathscr{NP}$. □

4. Can the candidate be in $\mathscr{P}$?

In this section, we further show that the candidate presented in the previous section is indeed in $\mathscr{NPI}$ under ETH.

Definition 4.1 (SE). A language $L \in SE$ if for any $x \in L$ there exists an algorithm to find a y so that $|y| \leqslant m(x)$ and $R(x, y)$ in time $\text{poly}(|x|)2^{\varepsilon m(x)}$ for every fixed ε, $1 > \varepsilon > 0$, where R is a polynomial time relation called the constraint, and m is a polynomial-time computable and polynomial bounded complexity parameter.

Definition 4.2 (SERF). The sub-exponential reduction family SERF from A_1 with parameter m_1 to A_2 with parameter m_2 is defined as a collection of Turing reduction $M_\varepsilon^{A_2}$, such that for each ε, $1 > \varepsilon > 0$:

(1) $M_\varepsilon^{A_2}(x)$ runs in time at most $\text{poly}(|x|)2^{\varepsilon m_1(x)}$.
(2) If $M_\varepsilon^{A_2}(x)$ queries A_2 with the input x', then $m_2(x') = \text{O}(m_1(x))$ and $|x'| = |x|^{\text{O}(1)}$.

If such a reduction family exists, A_1 is SERF-reducible to A_2. If each problem in $\mathscr{SNP}$ is SERF-reducible to a problem A, then A is $\mathscr{SNP}$-hard under SERF-reduction. And if A is also in $\mathscr{SNP}$ then we say A is $\mathscr{SNP}$-complete under SERF-reductions. Note that the SERF-reducibility is transitive, and, if (A_1, m_1) SERF-reduces to (A_2, m_2), and $(A_2, m_2) \in \text{SE}$, then $(A_1, m_1) \in \text{SE}$ [5].

Definition 4.3 (Strong many-one reduction). Let A_1 be a problem with complexity parameter m_1 and constraint R_1 and A_2 be a problem with complexity parameter m_2 and constraint R_2. A many-one reduction f from A_1 to A_2 is called a strong many-one reduction if $m_2(f(x)) = \text{O}(m_1(x))$. Strong many-one reduction is a special case of SERF-reduction [5].

Lemma 4.1. *3-SAT with complexity parameter n, the number of variables, is SERF-reducible to 3-SAT with complexity parameter m, the number of clauses* [5].

Lemma 4.2. *3-SAT is $\mathcal{SNP}$-complete under SERF-reductions, with either clauses or variables as the parameter* [5].

Definition 4.4 (3-ESAT). 3-ESAT is a variant of 3-SAT, satisfying that in any instance of 3-ESAT, say a formula Φ, the clause number is equal to the number of variables that appear in Φ.

Claim 3. *Given n ($n \geqslant 5$) variables, we can construct a satisfiable formula Φ in polynomial time of n, where Φ is an instance of 3-SAT and $|\Phi| \leqslant 2n$.*

Proof. We construct $2n$ 3-clauses with the form $x_i \vee x_j \vee x_k$, where $1 \leqslant i, j, k \leqslant n$, $i \neq j$, $i \neq k$, $j \neq k$. This can be done since there are $C_n^3 \geqslant 2n$ 3-clauses with such form. Then we select k, $1 \leqslant k \leqslant 2n$, 3-clauses to construct the formula Φ. Φ is satisfiable when all these n variables are assigned the value "true". □

Theorem 4.1. *3-ESAT is $\mathcal{SNP}$-hard under SERF-reductions, with either clauses or variables as the parameter. Consequently, 3-ESAT $\in SE$ implies $\mathcal{SNP} \subseteq SE$.*

Proof. According to Lemma 4.1, Lemma 4.2 and the definition of strong many-one reduction, we only need to show there exists a strong many-one reduction from 3-SAT with m (the clause number) as complexity parameter to 3-ESAT with m as complexity parameter.

For any given instance of 3-SAT, a formula Φ_0 which contains n_0 variables and m_0 clauses, we construct the many-one reduction, respectively, according to whether $m_0 > n_0$ or not.

Firstly, if $m_0 > n_0$, we add $\frac{3}{2}(m_0 - n_0)$ new variables and use them to construct a formula Φ_1 which contains $\frac{1}{2}(m_0 - n_0)$ clauses, in which each of all those $\frac{3}{2}(m_0 - n_0)$ new variables appears once and only once. This means that Φ_1 is always satisfiable. Let $\Phi = \Phi_1 \wedge \Phi_0$ then we get the instance of 3-ESAT since $m_0 + \frac{1}{2}(m_0 - n_0) = n_0 + \frac{3}{2}(m_0 - n_0)$, and Φ is satisfiable if and only if Φ_0 is satisfiable, and the reduction can be done in polynomial time of n_0.

Note that $m_0 + \frac{1}{2}(m_0 - n_0) < 2m_0$.

In the second case, we add n_1 new variables, where $n_1 = max\{n_0 - m_0, 5\}$ and construct a satisfiable formula Φ_1, with the size $(n_1 + n_0 - m_0)$. This can be done according to Claim 3 since $n_1 + n_0 - m_0 \leqslant 2n_1$. Then similar to the first case, let $\Phi = \Phi_1 \wedge \Phi_0$, we get the instance of 3-ESAT with parameter $n_1 + n_0$ and Φ is satisfiable if and only if Φ_0 is satisfiable. Thus, the reduction is done in polynomial time of n_0.

Note that

$$(n_1 + n_0 - m_0) + m_0 = \max\{2n_0 - m_0, 5 + n_0\} \leqslant \max\{5m_0, 3m_0 + 5\}.$$

Then according to the properties of SERF-reduction, the theorem does hold. □

From the above proof, it is also easy to see that 3-ESAT is also $\mathcal{NP}$-complete.

Definition 4.5 (ETH). Define s to be the infimum of $\{\delta$: there exists an $O(2^{\delta n})$ algorithm for solving 3-ESAT$\}$. Define ETH for 3-ESAT to be that: $s > 0$. In other words, 3-ESAT does not have sub-exponential time algorithm.

Note that this hypothesis is stronger than $\mathcal{NP} \neq \mathcal{P}$ but yet plausible according to both theoretical and practical arguments presented in Section 1. Under this assumption, we have the following result.

Theorem 4.2. *In the* $(2 + f(n))$*-SAT model, if the variables which appear in* Φ_2 *do not appear in* Φ_3*, and vice versa, then the* $(2 + (\log n)^k/n^2)$*-SAT is indeed in* $\mathcal{NPI}$ *under ETH for 3-ESAT,* $k \geqslant 2$.

Proof. Consider the special case of $(2 + (\log n)^k/n^2)$-SAT, where Φ_3 is an instance of 3-ESAT and $n_3 = m_3 = (\log n)^k$ and Φ_2 is always satisfiable. That is,

$$\frac{m_3}{m} = \frac{m_3}{m_2 + m_3} = \frac{(\log n)^k}{n^2} = \frac{m_3}{(n_2 + n_3)^2},$$

$$m_2 = (n_2 + n_3)^2 - m_3 \leqslant (n_2 + n_3)^2.$$

Note that $n_2 = n - n_3 = n - (\log n)^k$, $n_3 = (\log n)^k$, for sufficiently large n we get

$$(n_2 + n_3)^2 \leqslant \tfrac{3}{2}n_2^2 - \tfrac{3}{2}n_2.$$

This means the special case of $(2 + (\log n)^k/n^2)$-SAT indeed exists according to Claim 1.

Then for this special case of $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$, Φ_3 cannot be solved in polynomial time of n under ETH for 3-ESAT since there are $(\log n)^k$ variables in Φ_3, so does $\Phi = \Phi_2 \wedge \Phi_3$ since the variables which appear in Φ_2 do not appear in Φ_3, and vice versa.

Thus, $(2 + (\log n)^k/n^2)$-SAT is indeed not in $\mathcal{P}$ under ETH for 3-ESAT, $k \geqslant 2$, and according to theorem 3 the theorem does hold. □

The more general case of $(2 + (\log n)^k/n^2)$-SAT $(k \geqslant 2)$, where the variables which appear in Φ_2 may appear in Φ_3, and vice versa, is currently under investigation.

5. Remarks and conclusion

In this work, we study the boundary between $\mathcal{P}$ and $\mathcal{NPC}$ for the model of $(2 + p)$-SAT when p is considered as a function of n, the number of variables in the Boolean formula. The model allows us to obtain a natural problem in $\mathcal{NPI}$ under the ETH assumption. It is an interesting open problem whether this can be further shown to be in $\mathcal{NPI}$ under the weaker assumption $\mathcal{NP} \neq \mathcal{P}$.

Acknowledgements

The authors are grateful to the anonymous referees for their many valuable suggestions and constructive criticism that has improved former versions of this paper greatly. The authors are also grateful to Shirley Cheung for her valuable helps in forming this paper.

References

[1] W. Anderson, Solving problems in finite time, Nature 400 (1999) 115–116.
[2] R. Canetti, O. Goldreich, S. Goldwasser, S. Micali, Resettable zero-knowledge, in: Frances Yao (Ed.), Proceedings of the STOC'00, ACM Press, Portland, OR, USA, 2000, pp. 235–244.
[3] C. Dwork, M. Naor, Zaps and their applications, in: Proceedings of the FOCS'00, IEEE Computer Society Press, Redondo Beach, Canada, 2000, pp. 283–293.
[4] O. Goldreich, Introduction to complexity, Lecture Notes, Weizmann Institute, Israel, 1999, pp. 23–25, available from http://theory.lcs.mit.edu/~oded/.
[5] R. Impagliazzo, R. Paturi, Which problems have strongly exponential complexity?, in: Proceedings of the FOCS'98, IEEE Computer Society Press, Palo Alto, Canada, 1998, pp. 653–664.
[6] R. Impagliazzo, R. Paturi, Complexity of k-SAT, J. Comput. System Sci. 62 (2001) 367–375.
[7] S. Kirkpatrick, B. Selman, Critical behavior in the satisfiability of random Boolean expressions, Science 264 (1994) 1297–1301.
[8] R.E. Lander, On the structure of polynomial time reducibility, J. Assoc. Comput. Mach. 22 (1975) 155–171.
[9] S. Micali, M. Rabin, S. Vadhan, Verifiable random functions, in: Proceedings of the FOCS'99, IEEE Computer Society Press, New York, USA, 1999, pp. 120–130.
[10] S. Micali, L. Reyzin, Soundness in the public-key model, in: Joe Killian (Ed.), Proceedings of the Crypto'01, Lecture Notes in Computer Science, Vol. 2139, Springer, Berlin, 2001, pp. 542–565.
[11] R. Monasson, R. Zecchina, Tricritical points in random combinatorics: the $2+p$ SAT case, J. Phys. A 31 (1998) 9209–9217.
[12] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature 400 (1999) 133–137.
[13] H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, MA, 1994, pp. 329–332.
[14] U. Schoning, A probabilistic algorithm for k-SAT and constraint satisfaction problems, in: Proceedings of the FOCS'99, IEEE Computer Society Press, New York, USA, 1999, pp. 410–420.